DETAILED ACTION

In response to Amendments/Remarks filed 01/27/2010 and telephone interviewed on 04/07/2010, the examiner's amendment was authorized by attorney of record Chris McKenna, Attorney for Applicants.

- The Specification is currently amended.
- Claims 1, 6, 8, 11, 13, and 15 are currently amended.
- Claims 2-3, 7, 9-10, 14, and 16-18 are canceled.
- Claims 4, 5, and 12 were previously presented.

Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Information Disclosure Statement

A signed and dated copy of applicant's IDS, which was filed on 01/27/2010, is attached to this Office Action.

EXAMINER'S AMENDMENT

The application has been amended as follows:

In the Specification:

- Please replace paragraph that begins at Page 5, Line 20 and ends on Page 6, Line 2 with the following new paragraph:

The DSP 100 gathers content from a plurality of web servers 104 and uses a derivative server 110 to combine the content to produce a page 108 (or set of pages) for the client 102. (One client is shown, but in practice, the DSP 100 will maintain concurrent connections with thousands of clients 102, providing the described functionality for each.) The client 102 does not access each web server 104 directly, but rather obtains an assembled page 108 from the DSP 100 after sending it a request for a page. In many cases, the requested page does not exist on any one server 104, but rather is constructed by the derivative server 110 from multiple pages from different web servers 104. Each web server 104 is accessed by a virtual browser [[106]] 106a-106n ("106") belonging to the DSP 100. Since a typical browser such as Microsoft's Internet Explorer or Netscape's Navigator can easily require several megabytes of memory to execute, there is a high cost to simultaneously executing a large number of virtual browsers. For a DSP 100 to serve hundreds or thousands of clients 102 simultaneously, with each client 102 requiring a different dedicated virtual browser 106 for each site 104 forming part of that client's assembled page 108, the DSP 100 would have hundreds or thousands of virtual browsers simultaneously available, with the memory and processing power available to support them.

In the Claims:

- Please replace the entirety of the claims with the following claim set:

1. (Currently Amended) A computer-implemented method for efficiently parsing received data files, comprising:

receiving, by a virtual browser executing on a ~~device~~ server that is intermediary to a plurality of clients and a plurality of web servers, a data file from [[a]]one of the plurality of web servers, responsive to a request by ~~a client~~ one of the plurality of clients;

determining that the received data file comprises an object that is not cached on the server;

determining whether the object is currently being tracked;

retrieving~~, by the virtual browser,~~ a previously stored version of the data file and a syntax tree comprising nodes and tokens representing data within the previously stored version of the data file, the syntax tree including at least one static node;

comparing~~, by a comparison engine in communication with the virtual browser,~~ the previously stored version of the data file with the received data file ~~to identify~~ and identifying non-matching content in the received data file;

parsing~~, by a parsing engine of the virtual browser,~~ only the non-matching content of the received data file ~~to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file~~;

updating the syntax tree by replacing~~, by the virtual browser,~~ the at least one static node of the syntax tree with a new token;

creating~~, by the virtual browser,~~ a mapping from ~~each~~ the new token to ~~the at least one of the subtrees~~ a subtree in the syntax tree; and

~~transmitting, by the device, the syntax tree to the client responsive to the request~~ storing the updated syntax tree.

2.-3. (Canceled)

4. (Previously Presented) The computer-implemented method of claim 1 wherein the data file is a web page.

5. (Previously Presented) The computer-implemented method of claim 1 wherein the data file is an HTML file.

6. (Currently Amended) A method for efficiently parsing web pages, comprising:

receiving, by a virtual browser executing on a ~~device~~ server that is intermediary to a plurality of a clients and a plurality of web servers, ~~a first~~ an HTML page from [[a]] one of the plurality of web servers, responsive to a request by ~~a client~~ one of the plurality of clients;

determining that the received HTML page comprises an object that is not cached on the device server;

determining whether the object is currently being tracked;

retrieving~~, by the virtual browser,~~ a previously cached version of the HTML page and a syntax tree comprising nodes and tokens representing data within the ~~first~~ previously cached version of the HTML page, the syntax tree including at least one static node;

comparing~~, by a comparison engine in communication with the virtual browser,~~ the previously cached version of the HTML page with the received HTML page ~~to identify~~ and identifying non-matching content in the received HTML page;

parsing~~, by a parsing engine of the virtual browser,~~ only the non-matching content in the received HTML page ~~to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file~~;

updating the syntax tree by replacing~~, by the virtual browser,~~ the at least one static node of the syntax tree with a new token;

creating~~, by the virtual browser,~~ a mapping from ~~each~~ the new token to ~~the at least one of the subtrees~~ a subtree in the syntax tree; and

~~transmitting, by the device, the syntax tree to the client responsive to the request~~ storing the updated syntax tree and a most recent version of the HTML page.

7. (Canceled)

8. (Currently Amended) A method for efficiently parsing HTML pages, comprising:

receiving, by a virtual browser executing on a ~~device~~ server that is intermediary to a plurality of a clients and a plurality of web servers, ~~a first~~ an HTML page from [[a]]

one of the plurality of web servers, responsive to a request by ~~a client~~ one of the plurality of clients;

responsive to a determination that a previously cached version of the HTML page exists:

retrieving~~, by the virtual browser from a cache,~~ the previously cached version of the HTML page and a ~~first~~ syntax tree comprising nodes and tokens representing data within the ~~first~~ previously cached version of the HTML page, the ~~first~~ syntax tree including at least one static node;

comparing~~, by a comparison engine in communication with the virtual browser,~~ the previously cached version of the ~~first~~ HTML page with the received HTML page ~~to identify~~ and identifying non-matching content in the received HTML page;

parsing~~, by a parsing engine of the virtual browser,~~ only the non-matching content in the received HTML page ~~to form a subtree~~;

updating the syntax tree by replacing the at least one static node of the syntax tree with a new token; and

creating~~, by the virtual browser,~~ a mapping from [[a]]the new token ~~of the first tree to the subtree~~ to a subtree in the syntax tree;

responsive to a determination that [[the]]a previously cached version of the HTML page does not exist:

parsing~~, by the parsing engine of the virtual browser,~~ the received HTML page ~~to form~~ and building a ~~second~~ syntax tree comprising nodes and tokens representing ~~the non-matching~~ content of the received data file, the ~~second~~ syntax tree containing ~~at least one~~ only static nodes; and

storing the ~~second~~ syntax tree and the received HTML page in [[the]]a cache.

9.-10. (Cancelled)

11. (Currently Amended) A method for efficiently parsing received data files, comprising:

receiving, by a virtual browser executing on a ~~device~~ server that is intermediary to a plurality of clients and a plurality of web servers, a ~~first~~ data file from [[a]]one of the plurality of web servers, responsive to a request by ~~a client~~ one of the plurality of clients;

determining that the received data file comprises an object that is not cached on the server;

determining whether the object is currently being tracked;

~~retrieving a stored syntax tree from a cache, the stored syntax tree comprising nodes and tokens, representing data within the first data file and containing at least one static node and at least one token;~~

~~retrieving, by the virtual browser, a second data file from the cache, the second data file associated with the first data file;~~

retrieving a previously stored version of the data file and a syntax tree comprising nodes and tokens representing data within the previously stored version of the data file, the syntax tree including at least one static node;

comparing the previously stored version of the data file with the received data file and identifying~~, by a comparison engine in communication with the virtual browser,~~ non-matching content present only in the ~~first~~ received data file;

parsing~~, by a parsing engine of the virtual browser,~~ only the non-matching content of the first data file ~~to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file~~;[[and]]

updating the syntax tree by replacing the at least one static node of the syntax tree with a new token;

mapping~~, by the virtual browser, at least one of~~ the new token[[s]] to ~~at least one of the subtrees~~ a subtree in the syntax tree; and

storing the updated syntax tree.

12. (Previously Presented) The method of claim 11, further comprising:

responsive to identifying non-matching content present only in the ~~first~~ received file:

adding~~, by the virtual browser, at least one~~ the new token to the syntax tree.

13. (Currently Amended) A system for efficiently parsing input data from a plurality of content servers, comprising:

a server that is intermediary to a plurality of clients and a plurality of web servers, wherein the server performs the following functions:

~~a virtual browser, deployed on a device intermediary to a plurality of clients and a plurality of web servers, for retrieving content from content servers;~~

~~an identification engine, in communication with the virtual browser for identifying retrieved content;~~

~~a cache, in communication with the virtual browser, for storing retrieved content and syntax trees comprising nodes and tokens representing data within the retrieved content;~~

~~a comparison engine in communication with the virtual browser, for comparing retrieved content with stored content to identify non-matching content not stored in the cache; and~~

~~a parsing engine of the virtual browser for parsing only the non-matching content identified by the comparison engine, forming subtrees comprising nodes and tokens representing the non-matching content of the received data file and creating a mapping from new tokens to formed subtrees.~~

receiving a data file from one of the plurality of web servers, responsive to a request by one of the plurality of clients;

determining that the received data file comprises an object that is not cached on the server;

determining whether the object is currently being tracked;

retrieving a previously stored version of the data file and a syntax tree comprising nodes and tokens representing data within the previously stored version of the data file, the syntax tree including at least one static node;

comparing the previously stored version of the data file with the received data file and identifying non-matching content in the received data file;

parsing only the non-matching content of the received data file;

updating the syntax tree by replacing the at least one static node of the syntax tree with a new token;

creating a mapping from the new token to a subtree in the syntax tree; and

storing the updated syntax tree.

14. (Canceled)

15. (Currently Amended) ~~An intermediary~~ A system for efficiently parsing received data files transmitted between a client and a server, the ~~intermediary~~ system comprising:

a server that is intermediary to a plurality of clients and a plurality of web servers, wherein the server performs the following functions:

~~a cache storing a version of a data file received from a server and a syntax tree comprising nodes and tokens representing data within the data file, the tree including at least one static node;~~

~~a comparison engine comparing the stored version of the data file with the received data file to identify non-matching content in the received data file; and~~

~~a virtual browser, executing on a device intermediary deployed between a plurality of clients and a plurality of web servers, in communication with the comparison engine, retrieving the stored version of the data file and the syntax tree from the cache,~~

~~parsing only the non-matching content of the received data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file, replacing at least one static node of the syntax tree with a token, and creating a mapping from each token to one of the subtrees.~~

receiving a data file from one of the plurality of web servers, responsive to a request by one of the plurality of clients;

determining that the received data file comprises an object that is not cached on the server;

determining whether the object is currently being tracked;

retrieving a previously stored version of the data file and a syntax tree comprising nodes and tokens representing data within the previously stored version of the data file, the syntax tree including at least one static node;

comparing the previously stored version of the data file with the received data file and identifying non-matching content in the received data file;

parsing only the non-matching content of the received data file;

updating the syntax tree by replacing the at least one static node of the syntax tree with a new token;

creating a mapping from the new token to a subtree in the syntax tree; and

storing the updated syntax tree.

16.-18. (Canceled)

Allowable Subject Matter

The prior art made of record:

➢ US 20060242145A1	Krishnamurthy. et al	Filed	08/25/2000
➢ US006430624B1	Jamtgaard, et al.	Filed	02/14/2000
➢ US006718361B1	Basani, et al.	Filed	04/07/2000

❖ Claim(s) 1, 4-6, 8, 11-13, and 15 are allowed:

The following is a statement of reasons for the indication of allowable subject matter:

Interpreting the claims in light of the specification, Examiner finds the claimed invention is patentably distinct from the prior art of record, which set forth in the followings:

- ➢ ***Krishnamurthy*** teaches a method for tracking changes in pages by computing page differences; that is only transmitting the page difference to reconstruct the new page. This is generally discloses at Para [0076] and illustrates in Fig(s) 7 and 8 of Krisnamurhty.
- ➢ ***Jamtgaard*** enabling the web content to be fed into a virtual browser. This virtual browser provides the important functionality of proxying javascript and cookies for the target devices. This is generally discloses at Col. 10 Lines 21-47 and illustrates in Fig 7 of Jamtgaard.

- ***Basani*** discloses a load-balancing process directly determine whether a particular server has the most recent version of content. This is generally discloses at Col. 4 Lines 45-57 and illustrates in Fig 1 of Basani.

Under the broadest reasonable interpretation of the claimed limitation which is consistence with the Applicant's Specification. The prior art cited above fails to teach all of the Applicant's claimed limitation. In particularly, the claimed invention advantageously provides a finer level of detail that enables a ***serve*** that can run concurrently by DSP (Derivative Service Provider); determining which portions of a page are ***static, and which are dynamic***; then ***parsing only the dynamic content***, in other word, comparing/parsing HTML pages to identify ***only the non-matching content in the received data file*** by the virtual browser (e.g., not by client computer or by web server); then modifying, by the virtual browser service, the abstract syntax tree to comprise a token mapped to the subtree; and transmitting***, by the server service***, ***the modified abstract syntax tree*** to a client responsive to a request by the client for the web page. This would enable the derivative services overcome the excessive memory and processor requirements for parsing html content; since much of the content in a given content page remains static over time and only the dynamic content are reparsed [see currently amended claims 1, 6, 8, 11, 13, and 15-16 and the specification @ page 7 lines 1-7; @ the Abstract and illustrates in Fig 1].

The Examiner asserts that the claims overcome the prior art of record as describes above when the limitations are read in combination with the respective claimed limitations in their entirety.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Mon through Fri 8AM - 5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on (571)272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Quoc A. Tran/
Examiner, Art Unit 2176

/DOUG HUTTON/
Supervisory Patent Examiner, Art Unit 2176